(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平11－85710

(43)公開日　平成11年(1999)３月30日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　15/16	３７０	Ｇ０６Ｆ　15/16	３７０Ｍ

審査請求　未請求　請求項の数６　ＯＬ　（全 12 頁）

(21)出願番号　特願平9－250249

(22)出願日　平成９年(1997)９月16日

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者　浅野　滋博
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

(72)発明者　金井　達徳
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

(72)発明者　菅野　伸一
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内

(74)代理人　弁理士　外川　英明

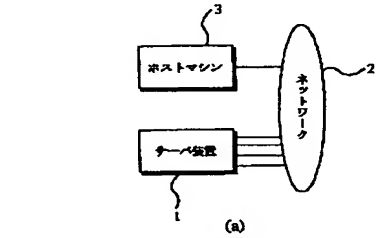
最終頁に続く

(54)【発明の名称】　サーバ装置およびファイル管理方法

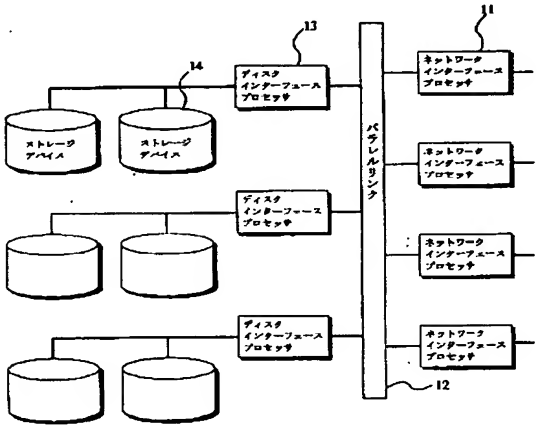
(57)【要約】

【解決課題】　本発明は、管理が簡単でかつコストパフォーマンスに優れたサーバを提供することを目的とするとともに、階層キャッシュシステムにおける効率のよいファイル管理方法をも提供することを目的とする。

【解決手段】　本発明は、少なくとも第１のプロセッサ及び第１の記憶手段を有する複数の第１の装置（１１）と、前記複数の第１の装置と転送手段（１２）を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置（１３）と、前記複数の第２の装置の各々に接続された第３の記憶手段（１４）とを具備し、前記第１の装置に対して与えられるネットワークからの要求に基づく処理を、所定の条件に応じて、前記第１の記憶手段、前記第２の記憶手段または前記第３の記憶手段のうちのいずれかの記憶手段に対して行うことを特徴とするサーバ装置である。



(a)



(b)

1

【特許請求の範囲】

【請求項１】　少なくとも第１のプロセッサ及び第１の記憶手段を有する複数の第１の装置と、
前記複数の第１の装置と転送手段を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置と、
前記複数の第２の装置の各々に接続された第３の記憶手段とを具備し、
前記第１の装置に対して与えられるネットワークからの要求に基づく処理を、所定の条件に応じて、前記第１の記憶手段、前記第２の記憶手段または前記第３の記憶手段のうちのいずれかの記憶手段に対して行うことを特徴とするサーバ装置。

【請求項２】　ネットワークから要求を与えられる前記第１の装置は、該要求に関連する所定の情報に基づいて、前記第２の装置を選択することを特徴とする請求項１記載のサーバ装置。

【請求項３】　前記第３の記憶手段から前記第１の装置に所定のデータが送出される際に、所定の計算を行い、該得られた計算結果を該所定のデータとともに、前記第１の記憶手段に記憶することを特徴とする請求項１記載のサーバ装置。

【請求項４】　少なくとも第１のプロセッサ及び第１の記憶手段を有する複数の第１の装置と、
前記複数の第１の装置と転送手段を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置と、
前記複数の第２の装置の各々に接続された第３の記憶手段とを具備し、
前記第１の記憶手段に第１のデータが記憶され、前記第２の記憶手段に該第１のデータに対応する第２のデータが記憶されている場合に、前記第１の記憶手段から該第１のデータが追い出されるときは、前記第２の記憶手段に記憶された第２のデータの追い出し順位を他のデータの追い出し順位よりも低く設定するための処理を行うことを特徴とするサーバ装置。

【請求項５】　少なくとも第１のプロセッサ及び第１の記憶手段を有する複数の第１の装置と、
前記複数の第１の装置と転送手段を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置と、
前記複数の第２の装置の各々に接続された第３の記憶手段とを具備するサーバ装置におけるファイル管理方法であって、
前記第１の装置に対して与えられるネットワークからの要求に基づく処理を、所定の条件に応じて、前記第１の記憶手段、前記第２の記憶手段または前記第３の記憶手段のうちのいずれかの記憶手段に対して行うことを特徴とするファイル管理方法。

【請求項６】　少なくとも第１のプロセッサ及び第１の

2

記憶手段を有する複数の第１の装置と、
前記複数の第１の装置と転送手段を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置と、
前記複数の第２の装置の各々に接続された第３の記憶手段とを具備するサーバ装置におけるファイル管理方法であって、
前記第１の記憶手段に第１のデータが記憶され、前記第２の記憶手段に該第１のデータに対応する第２のデータが記憶されている場合に、前記第１の記憶手段から該第１のデータが追い出されるときは、前記第２の記憶手段に記憶された第２のデータの追い出し順位を他のデータの追い出し順位よりも低く設定するための処理を行うことを特徴とするファイル管理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、サーバ装置およびファイル管理方法に関するものである。

【０００２】

【従来の技術】インターネットの発達により情報の流通の速度が飛躍的に高まり、ますます多くの人がインターネットにアクセスするようになっている。情報を提供する手段としてはWorld Wide Web（以下「Web」という）によるものが一般的であり、プロトコルはＨＴＴＰおよびＴＣＰ／ＩＰが使用される。情報を多くの人に提供するためには、ネットワークのバンド幅の拡大だけでなく、情報を蓄積したサーバがネットワークに送り出す能力の拡大も同時に必要である。

【０００３】サーバに必要な機能は、主として、ストレージデバイスに蓄積された情報をネットワークに送り出す機能であるが、ストレージデバイスに蓄積された情報のフォーマットとネットワークに送り出す情報のフォーマットが異なるために、フォーマットの変換を行わなければならない。また、ＨＴＴＰやＴＣＰ／ＩＰで通信するためにプロトコルを実行する必要がある。

【０００４】従って、これらの処理を１台のプロセッサで行っていたのではプロセッサにかかる負荷が増大し、性能向上のボトルネックとなる。また、前述したフォーマット変換のためにプロセッサがメモリ内で情報をコピーすることが頻繁に起こるので大きなメモリバンド幅が必要になる。

【０００５】これらのボトルネックを緩和するためにプロセッサおよびメモリを複数結合するマルチプロセッサ方式のサーバが知られている。例えば、分散ＰＣ方式は、パフォーマンスを向上させるために、ネットワーク上に複数のＷｅｂサーバを設置したものである。しかしながら、この分散ＰＣ方式では、複数のＷｅｂサーバのそれぞれにファイルを分割するためファイル管理が煩雑であるとともに、どのＷｅｂサーバにアクセスするかという制御が必要であり、このため複雑な管理が必要にな

る。また、あるＷｅｂサーバに蓄積されているファイルにアクセスが偏ると、そのＷｅｂサーバの負荷が増えてしまうという問題もあった。

【０００６】また、共有メモリマルチプロセッサ方式においては、ＣＰＵのキャッシュメモリにある程度のヒット率が見込まれることを前提に、それにあわせたバスおよびメモリのバンド幅を用意する。しかしながら、共有メモリマルチプロセッサをＷｅｂサーバに使用した場合には、ＣＰＵのキャッシュにはほとんどヒットしないのでこれがボトルネックとなり、高価で高速なＣＰＵの能力は有効に活用することができなかった。

【０００７】さらに、フロントエンド＋ファイルサーバ方式は、フロントエンドでＨＴＴＰを実行し、必要なファイルをバックエンドネットワークを経由してファイルサーバに要求するというものである（"Application of NFS Servers to Strategic Internet/ Intranet Website Design " Technical Report 13.Version 1.0.July1996. Auspex Systems. Inc. Santa Clara. California）。

【０００８】しかしながら、この方式では、フロントエンドとファイルサーバの間の通信がバックエンドネットワークに委ねられている。そのため、バックエンドネットワークは、一般にＮＦＳなど標準の通信規約により行われ、この通信を行うためにフロントエンド、ファイルサーバともにＣＰＵの負荷が増大することが問題となる。

【０００９】また、機能分散マルチプロセッサ方式は、ネットワークコントローラ、ファイルコントローラおよびストレージコントローラが、共有バス上でシステムメモリを共有する構造を有しているものである（ＵＳＰ５，３５５，４５３）。

【００１０】しかしながら、この方式では、複数のネットワークコントローラの読み出しが、全てシステムメモリに集中するので、共有バスおよびシステムメモリのボトルネックになりやすいという問題があった。

【００１１】また、疎結合マルチプロセッサ方式は、ハイパーキューブ等で結合されたプロセッサ間結合網を持った疎結合のマルチプロセッサをＷｅｂサーバとして使用する（"A Scalable and Highly Available Web Server" Proceedings of the IEEE Computer Conference (COMPCON). Santa Clara. March. 1996.）。

【００１２】しかしながら、この方式では、プロセッサ間結合網のコストが非常に高いことからシステム全体の価格を押し上げるという問題があった。ところで、ＴＣＰ／ＩＰは、ネットワークに広く使用されているプロトコルであるが、ＴＣＰ／ＩＰのパケットをネットワークに送り出す場合は、パケットに対してパリティを計算し、ヘッダ部に付加する必要がある。しかるに、従来は、パリティの計算はＣＰＵによって送り出すデータを読み出して計算を行っていたが、この計算はメモリバンド幅にもＣＰＵにも負荷をかけるので望ましくない。

【００１３】また、複数のキャッシュ階層の管理においては、第１のキャッシュメモリの内容を第２のキャッシュメモリが全て包含するというマルチレベルインクルージョンという性質が満たされているのが一般的であった。この方法は、第１のキャッシュメモリの速度より第２のキャッシュメモリの速度が遅く、第２のキャッシュメモリはビット単価が安いので容量が大きいという前提において効率的である。

【００１４】しかしながら、第１のキャッシュメモリ手段、第２のキャッシュメモリ手段の速度の差は大きくないような構成において、キャッシュミスした場合、ディスクアクセスにより多大な遅延時間をともなう。この場合、キャッシュの実効的な容量を増大させるのが重要であるにもかかわらず、従来のキャッシュ管理方式においては、第１のキャッシュ手段と、第２のキャッシュ手段とに同じ内容が入っているのでメモリ容量にむだが生じることになっていた。

【００１５】

【発明が解決しようとする課題】上記従来のサーバの構成では、その管理が煩雑であったり、コストパフォーマンスが低いという問題があった。また、パリティの計算はＣＰＵによって送り出すデータを読み出して計算を行っていたが、このような計算はメモリバンド幅にもＣＰＵにも負荷をかけるので望ましくないかった。

【００１６】さらに、従来のキャッシュ管理方式においては、第１のキャッシュ手段と、第２のキャッシュ手段とに同じ内容が入っているのでメモリ容量にむだが生じることになっていた。

【００１７】そこで、本発明は、管理が簡単でかつコストパフォーマンスに優れたサーバを提供することを目的とする。また、本発明は、効率のよいネットワークへの送出手段を提供することをも目的とする。さらに、本発明は、階層キャッシュシステムにおける効率のよい記憶領域管理方式をも提供することを目的とする。

【００１８】

【課題を解決するための手段】請求項１に係る発明は、少なくとも第１のプロセッサ及び第１の記憶手段を有する複数の第１の装置（ネットワークインターフェースプロセッサ）と、前記複数の第１の装置と転送手段（内部バス）を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置（ディスクインターフェースプロセッサ）と、前記複数の第２の装置の各々に接続された第３の記憶手段（ストレージデバイス）とを具備し、前記第１の装置に対して与えられるネットワークからの要求に基づく処理を、所定の条件に応じて、前記第１の記憶手段、前記第２の記憶手段または前記第３の記憶手段のうちのいずれかの記憶手段に対して行うことを特徴とするサーバ装置である。

【００１９】また、請求項２に係る発明は、ネットワー

クから要求を与えられる前記第１の装置は、該要求に関連する所定の情報に基づいて、前記第２の装置を選択することを特徴とする。

【００２０】さらに、請求項３に係る発明は、前記第３の記憶手段から前記第１の装置に所定のデータが送出される際に、所定のパリティ計算を行い、該得られた計算結果を該所定のデータとともに、前記第１の記憶手段に記憶することを特徴とする。

【００２１】また、請求項４に係る発明は、少なくとも第１のプロセッサ及び第１の記憶手段を有する複数の第１の装置と、前記複数の第１の装置と転送手段を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置と、前記複数の第２の装置の各々に接続された第３の記憶手段とを具備し、前記第１の記憶手段に第１のデータが記憶され、前記第２の記憶手段に該第１のデータに対応する第２のデータが記憶されている場合に、前記第１の記憶手段から該第１のデータが追い出される（実際に追い出す場合のほか、データへのアクセスの無効化等を含む。）ときは、前記第２の記憶手段に記憶された第２のデータの追い出し順位を他のデータの追い出し順位よりも低く設定するための処理を行うことを特徴とするサーバ装置である。

【００２２】また、請求項５に係る発明は、少なくとも第１のプロセッサ及び第１の記憶手段を有する複数の第１の装置と、前記複数の第１の装置と転送手段を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置と、前記複数の第２の装置の各々に接続された第３の記憶手段とを具備するサーバ装置におけるファイル管理方法であって、前記第１の装置に対して与えられるネットワークからの要求に基づく処理を、所定の条件に応じて、前記第１の記憶手段、前記第２の記憶手段または前記第３の記憶手段のうちのいずれかの記憶手段に対して行うことを特徴とするファイル管理方法である。

【００２３】さらに、請求項６に係る発明は、少なくとも第１のプロセッサ及び第１の記憶手段を有する複数の第１の装置と、前記複数の第１の装置と転送手段を介して接続される、少なくとも第２のプロセッサ及び第２の記憶手段を有する複数の第２の装置と、前記複数の第２の装置の各々に接続された第３の記憶手段とを具備するサーバ装置におけるファイル管理方法であって、前記第１の記憶手段に第１のデータが記憶され、前記第２の記憶手段に該第１のデータに対応する第２のデータが記憶されている場合に、前記第１の記憶手段から該第１のデータが追い出されるときは、前記第２の記憶手段に記憶された第２のデータの追い出し順位を他のデータの追い出し順位よりも低く設定するための処理を行うことを特徴とするファイル管理方法である。

【００２４】本発明によれば、複数用意されたＮＩＰ間で通信プロトコルの処理を分散して行うことにより高い性能を得ることができる。また、ＮＩＰに用意された第１のバッファメモリ手段をキャッシュとして使用することによりサーバの内部バスのトラフィックを減少させることができ、スケーラビリティが得られるとともに通信のレーテンシを減少させる効果がある。ＤＩＰを複数用意することにより、ＤＩＰの負荷を分散し、ファイルシステムの処理能力を高めている。また、ＤＩＰに用意された第２のバッファメモリ手段がキャッシュとして働くことによりストレージデバイスへの負荷を低減している。

【００２５】また、本発明のパリティ計算手段を用いればＣＰＵに負荷をかけずにネットワークにデータを送り出すことが可能となる。さらに、本発明のキャッシュメモリ管理方式を用いれば、キャッシュメモリをより効率的に使えるので、少ないメモリ容量でパフォーマンスの増加が見込まれる。以上のように、本発明によれば、コストパフォーマンスがよく、スケーラブルなＷｅｂ等のサーバを構成することができる。

【００２６】

【発明の実施の形態】以下、図面を参照しながら本発明の実施例を説明する。図１（ａ）は、本発明の実施形態に係るサーバ装置を用いたシステムの概略構成を示す図である。同図に示すように、本発明に係るサーバ装置１は、ネットワーク２を介してホストマシン３と接続されている。なお、同図において、ホストマシン３は１つのみ接続されているが、特にこれにこだわる必要はなく、複数存在していてもよい。

【００２７】サーバ装置１は、後述するように、ファイルから読み出したデータをネットワークに送出する機能が実現される。また、サーバ装置１は、ホストプロセッサ（図示せず）が設けられており、当該ホストプロセッサが処理すべきものは、当該ホストプロセッサによって処理される。例えば、ＷｅｂサーバにおけるＣＧＩの実行は、ホストプロセッサが処理する。

【００２８】本実施形態においては、ホストプロセッサとサーバ装置との関係を明確にするために、ＣＧＩ実行を他のファイルからのアクセスと区別する方法について説明する。一般に、ＣＧＩ実行は、サーバ装置の所定のディレクトリ下で行われる。ＣＧＩ実行が行われるディレクトリを予めサーバ装置に設定しておき、このディレクトリの下にあるアクセスに対しては、ホストプロセッサにリダイレクトされるように設定しておく。このような設定は、例えば、環境ファイル等により行うことができる。

【００２９】図１（ｂ）は、本発明の実施形態に係るサーバ装置の構成を示す図である。同図に示すように、サーバ装置１は、ネットワーク２に接続されるネットワークインターフェースプロセッサ２１（ＮＩＰ：Network Interface Processor、以下「ＮＩＰ」ということもある。）が複数設けられている。ネットワークは、例え

ば、Ethernetが考えられるが、ＡＴＭ等、他のネットワークでもかまわない。

【００３０】各ＮＩＰ２１は、サーバ装置１の内部伝送路（例えば、いわゆる内部バス）であるパラレルリンク１２に接続されている。この内部バスは、パラレルリンクに限るものではなく、例えば、シリアルリンク、スイッチ等があげられる。また、このパラレルリンク１２には、ディスクインターフェースプロセッサ１３（ＤＩＰ：Disk Interface Processor、以下「ＤＩＰ」ということもある。）が少なくとも１つ以上接続されており、このＤＩＰにはストレージデバイス１４が設けられている。ストレージデバイス１４は、例えば、磁気ディスクドライブやディスクドライブ等により実現される。

【００３１】図２は、ネットワークインターフェースの構成を示す図である。同図に示すように、ＮＩＰ１１は、プロセッサ２１、ＮＩＰローカルメモリ２３、コード用のＲＯＭ２３、ネットワークコントローラ２４およびパラレルリンクインタフェース２５から構成される。

【００３２】プロセッサ２１が実行するプログラムコードは、あらかじめＲＯＭに書込まれている。一般に、ＲＯＭのアクセス速度は低速であるので、ＲＯＭの内容はブートストラッププログラムによりＮＩＰローカルメモリ２３にコピーされる。

【００３３】ＮＩＰ１１内の上記各ユニットは、例えば、ＰＣＩバス２６等の標準バスにより接続されている。なお、ＮＩＰの構成は、同図に限られるものではなく、プロセッサ２１、ＮＩＰローカルメモリ２２およびＰＣＩバスがブリッジチップ（図示せず）と呼ばれるＡＳＩＣにより接続されるようにしてもよい。ＮＩＰのブリッジチップはプロセッサ内のキャッシュメモリ（ＮＩＰローカルメモリ内のキャッシュとは異なる。）をスヌープする機能を持つ。スヌープ機構によってＤＭＡ転送がプロセッサのキャッシュメモリを更新し、一貫性を保つことができる。

【００３４】次に、ＮＩＰ１１の動作について、さらに詳細に説明する。ネットワークからのサーバ装置１に対する要求は、ＮＩＰ１１に伝えられる。プロトコルは、ＴＣＰ／ＩＰおよびＨＴＴＰが使用されるのが一般的であるがこれにこだわるものではない。この要求は、ＮＩＰ１１内のプロセッサ２１が所定のプログラム命令を実行することによって、解釈される。

【００３５】ＮＩＰ１１内のプロセッサ２１は、この要求がホストプロセッサにより実行されるべきものか否かを判断し、ホストプロセッサにより実行されるものであると判断した場合には、該要求をホストプロセッサにリダイレクトする。

【００３６】一方、サーバ装置１が実行すべき要求だと判断した場合には、要求されたデータがＮＩＰ１１内のキャッシュメモリに存在するか否かの判断を行う。本実施形態においては、ローカルメモリ２２がキャッシュメモリとして機能するが、例えば、プロセッサ２１内に設けられる１次キャッシュメモリを含むものであってもよい。

【００３７】ここで、所定のデータがキャッシュメモリ内に存在するか否か、すなわち、キャッシュメモリの検索について説明する。図３は、キャッシュエントリの構造を示す図である。同図に示すように、ローカルメモリ内に展開されているキャッシュの内容は、タグ部３１とデータ部３２とからなる各エントリが、ＬＲＵポインタ３１１によってリンクされており、ＬＲＵＬＡＳＴ３１２が最も古くにアクセスされたエントリを、ＬＲＵＴＯＰ３１３は最も新しくアクセスされたエントリを示している。このリンク構造は、アクセスがあるごとに更新される。ファイル名フィールド３１２は、エントリが保持しているデータ（ファイル）のファイル名を保持し、タグとして用いらる。ＴＴＬ（Time To Live）フィールド３１５は、キュッシュされたデータの有効時間を示す。すなわち、もし、ＴＴＬフィールド３１５の値（ＴＴＬ値）が現在の時刻と比較して有効期限がきれていると判断される場合には、たとえキャッシュにヒットしてもそのキャッシュの内容は有効でないとみなされる。このＴＴＬフィールド３１５は、ファイルの属性として付加されれており、ＤＩＰ１３からＮＩＰに転送される際に、ＴＴＬ値も同時に転送される。データ長フィールド３１６は、エントリのデータの長さを示し、データへのポインタフィールド３１７は、データの先頭アドレスを示す。

【００３８】データ部３２は、一定の長さの領域（例えば４Kbyte）に分割されたデータがポインタでリンクされた構造をしており、データ領域はフリーリストで管理される。これは、キャッシュされるデータは可変長なので、データ領域の管理を容易にするためである。

【００３９】以上のように、ローカルメモリ２２内に展開されたキャッシュの検索は、ＬＲＵＬＡＳＴ３１２から順にＬＲＵポインタ３１１をたどりながらＬＲＵを比較していくことによって行われる。検索を高速化する手法としては、ハッシング等が知られているが、これに限るものではない。

【００４０】なお、キャッシュメモリの容量には限界があるので、キャッシュに新しい内容をいれるときにはキャッシュ内の内容を追い出す必要がある。これにはＬＲＵアルゴリズム等を使用して、追い出すべきエントリーを決定することができる。追い出すエントリーが決定されると、ＮＩＰ１１はＤＩＰ１３に対して、どのエントリーが追い出されるかを通知する。以上がキャッシュの検索の説明である。

【００４１】プロセッサ２１が、要求されたデータをローカルメモリ２２のキャッシュから転送可能であると判断した場合には、プロセッサ２１は、ネットワーク送出に必要なヘッダ等の情報を構成し、ローカルメモリ２２

内に書き込む。さらにプロセッサ２１は、ネットワークコントローラ２４のDMA（図示せず）をセットしてローカルメモリ２２からネットワークコントローラ２４に転送し、ネットワークコントローラ２４が当該情報をネットワークに送出する。

【００４２】構成されるヘッダ等の情報は、Ｗｅｂサーバの場合、ＨＴＴＰプロトコルのヘッダ、ＴＣＰのヘッダおよびＩＰのヘッダが含まれる。ネットワークコントローラは、例えば、ＤＥＣ社のＤＥＣ２１１４０等があげられる。このチップは、ＰＣＩインタフェースとＤＭＡコントローラを備えている。

【００４３】プロセッサ２１が、要求されたデータをローカルメモリ２２から供給できないと判断した場合には、プロセッサ２１は、ＤＩＰ１３に対して要求を行う。要求は、内部バス（パラレルリンク）１２を介して電送される。要求は、必要なファイル情報の他、ＤＩＰ１３からＮＩＰ１１にデータを転送する際のＮＩＰ１１側のバッファメモリのアドレスが含まれる。

【００４４】ＮＩＰ１１のプロセッサ２１は、該要求を複数のＤＩＰのうちのどのＤＩＰに送出するかを決定する。本実施形態においては、ファイル名に基づいてハッシュ値を求め、該ハッシュ値に基づいて決定される。例えば、ＤＩＰ１３が３個設けられている場合、ハッシュ関数としてファイル名のキャラクタコードの和のｍｏｄ３をとったものが使用される。ＮＩＰ１１からＤＩＰ１３への要求の他、上述したようなどのエントリーを追い出したかという情報をＮＩＰ１１からＤＩＰ１３に通知する場合にも、同様の方法でＤＩＰ１３を決定する。

【００４５】図４は、パラレルリンクインタフェース２５の構成を示す図である。同図に示すように、パラレルリンクインターフェース２５は、パラレルリンク制御部４１、パリティ計算部４２、Ｒｅｃｅｉｖｅパリティバッファ４３、Ｒｅｃｅｉｖｅバッファ４４、Ｓｅｎｄバッファ４５、ＲｅｃｅｉｖｅＤＭＡ４６、ＳｅｎｄＤＭＡ４７およびＰＣＩインタフェース４８より構成される。

【００４６】パラレルリンク制御部４１は、ＤＩＰからの通信のうち、宛先が該当するＮＩＰに対する通信（メッセージ）のみを通過させ、パリティ計算部４２に送出する。パリティ計算部４２は、パリティの計算を行い、一定のワード数ごとにその結果をＲｅｃｅｉｖｅパリティバッファ４３に書き込む。Ｒｅｃｅｉｖｅパリティバッファ４３は、例えば、２５６ワードに１回の書き込みでは２５６ワード転送毎に書き込まれる。ここでのワードは、ＴＣＰのパリティの形式にあわせて１６ビットで１ワードとする。

【００４７】ＲｅｃｅｉｖｅＤＭＡ４６は、Ｒｅｃｅｉｖｅパリティバッファ４３に書込まれたデータを、ＮＩＰローカルメモリ２２に転送する。ＴＣＰパケットをＮＩＰ１１のプロセッサ２１が作成する場合、Ｒｅｃｅｉｖｅパリティバッファ４３から転送されたパリティデータを使用する。例えば、ＴＣＰパケットが１０２４ワードのとき、その１０２４ワードに対応する４ワードのパリティデータからパケットのパリティを作成する。１０２４ワードがパリティデータの２５６ワードのアライメントにあわない場合には、アライメントにあわない端数部分のパリティを計算する必要がある。Ｒｅｃｅｉｖｅパリティバッファ４３を設けることによって、ＮＩＰ１１のプロセッサ２１によるパリティ計算の負荷を著しく減少することができるようになる。また、２５６ワード毎にパリティを付加することにより、ＴＣＰパケットのサイズに柔軟に対応することができる。

【００４８】ＤＩＰに対する要求は、ＮＩＰ１１のプロセッサ２１によりコマンドが構成され、ＳｅｎｄＤＭＡ４７によりＳｅｎｄバッファ４５に転送され、さらにパラレルリンク１２に送出される。

【００４９】図５は、ＮＩＰ１１にネットワークから要求がきた場合の処理を説明するための図である。同図において、まず、ネットワークコントローラ２４を介してネットワークから要求を受け付けると（ＳＴＥＰ５１）、ＣＧＩか否かが判断される（ＳＴＥＰ５２）。要求がＣＧＩであると判断された場合には、ＮＩＰ１１のプロセッサ２１は、ＣＧＩを実行するように設定されていないため、ホストマシンに処理を要求する（ＳＴＥＰ５３）。一方、ＣＧＩでないと判断された場合には、次に、ＮＦＳであるか否かが判断される（ＳＴＥＰ５４）。

【００５０】ＮＦＳであると判断された場合には、要求を送出するＤＩＰを決定し（ＳＴＥＰ５５）、該決定されたＤＩＰにＮＦＳを要求する（ＳＴＥＰ５６）。一方、ＮＦＳでないと判断された場合には、上述のキャッシュの検索を行う（ＳＴＥＰ５７）。キャッシュを検索することによりヒットした場合には、ＴＴＬをチェックし（ＳＴＥＰ５８、５９）、有効期限内か否かを判断する（ＳＴＥＰ５１０）。ＴＴＬ値が有効期限内であると判断された場合には、ＮＩＰ１１のＮＩＰローカルメモリ２２からデータをネットワークに送出する（ＳＴＥＰ５１１）。一方、有効期限内でないと判断された場合には、要求を送るＤＩＰを決定し（ＳＴＥＰ５１２）、該決定されたＤＩＰに要求を送出する。

【００５１】図６は、ＮＩＰ１１がＤＩＰ１３からの応答を受け付けた場合の処理を説明するための図である。同図において、ＤＩＰから応答を受け付けると（ＳＴＥＰ６１）、ＮＦＳからの応答である否かが判断される（ＳＴＥＰ６２）。ＮＦＳからの応答であると判断された場合には、そのままネットワークに応答を返す（ＳＴＥＰ６３）。ＮＦＳからの応答でない場合に、要求したデータが帰ってきた場合には、ＮＩＰ１１は、ＮＩＰローカルメモリ２２内のキャッシュがいっぱいであるか否かを判断する（ＳＴＥＰ６４）。キャッシュがいっぱい

である場合には、追い出すべきエントリを決定し、該エントリを追い出すとともに（ＳＴＥＰ６５）、追い出したエントリをＤＩＰに通知する（ＳＴＥＰ６６）。キャッシュがいっぱいでないと判断された場合、またはエントリの追い出し処理を行った後、ＮＩＰ１１は、キャッシュに該送られてきたデータをキャッシュに書き込むとともに、ネットワークに送出する（ＳＴＥＰ６７）。

【００５２】次に、ＤＩＰ１３の詳細について説明する。図７は、ＤＩＰ１３の構成を示す図である。同図に示すように、ＤＩＰ１３は、プロセッサ７１、ＤＩＰローカルメモリ７２、コード用ＲＯＭ７３、ＳＣＳＩインタフェース７４、パラレルリンクインタフェース７５およびファイルバッファメモリ（図示せず）から構成される。

【００５３】ＤＩＰ１３内の上記各ユニットは、ＰＣＩバス等の標準バスにより接続されている。なお、図７に示す接続方法のほか、プロセッサ７１とＤＩＰローカルメモリ７３およびＰＣＩバス１１を接続するために、ブリッジチップと呼ばれるＡＳＩＣを用いてもよい。

【００５４】図８は、ＤＩＰ１３のパラレルリンクインタフェース７５の構成を示す図である。同図に示すように、パラレルリンクインターフェース７５は、パラレルリンク制御部８１、Ｒｅｃｅｉｖｅバッファ８２、ＳｅｎｄバッファＤＭＡ８３、ＲｅｃｅｉｖｅＤＭＡ８４、ＳｅｎｄＤＭＡ８５およびＰＣＩインタフェース８６より構成される。

【００５５】パラレルリンク制御部４１は、ＮＩＰ１１からの通信のうち、宛先が該当するＤＩＰに対する通信（メッセージ）のみを通過させ、Ｒｅｃｅｉｖｅバッファ８２に送出する。

【００５６】ＲｅｃｅｉｖｅＤＭＡ８４は、Ｒｅｃｅｉｖｅバッファ４２に書込まれたデータを、ＰＣＩインターフェース８６を介して、ＤＩＰローカルメモリ７２に転送する。

【００５７】ＮＩＰ１１に対する応答は、ＤＩＰ１２のプロセッサ７１によりコマンドが構成され、ＳｅｎｄＤＭＡ８５によりＳｅｎｄバッファ８３に転送され、さらにパラレルリンク８１に送出される。

【００５８】ＤＩＰ１３は、パラレルリンクインタフェース７５を介して、ＮＩＰ１１からのデータの要求またはキャッシュエントリの追い出し通知を受け取る。ＤＩＰ１３は、ＮＩＰ１１からのデータ要求を受け取ると、ＤＩＰローカルメモリ７２に該要求されたデータが存在するか否かを調べる。ＤＩＰローカルメモリもおいても、図３に示したＮＩＰローカルメモリと同様に、キャッシュとして機能させるための各種フィールドを有する。

【００５９】図９は、ＤＩＰの動作を説明するための図である。同図において、ＤＩＰ１３は、いづれかのＮＩＰ１１から通知を受け取ると（ＳＴＥＰ９１）、該通知

が追い出し通知であるか否かを判断する（ＳＴＥＰ９２）。該通知が追い出し通知であると判断された場合には、キャッシュに該当するエントリを検索する（ＳＴＥＰ９３）。検索の結果、該当するエントリがあるか否かを判断し（ＳＴＥＰ９４）、ないと判断された場合には、特になにも行わない。一方、該当するエントリがあると判断された場合には、該当するエントリをＬＲＵＴＯＰにつなぎ直す（ＳＴＥＰ９５）。これは、ＤＩＰローカルメモリ７２のキャッシュのエントリを追い出す場合に、ＮＩＰローカルメモリのキャッシュから追い出されたエントリであるか否かを考慮するものである。

【００６０】すなわち、ＮＩＰローカルメモリのあるエントリが追い出し通知を受けた場合に、そのエントリのコピーがＤＩＰローカルメモリに存在する場合には、ＤＩＰローカルメモリの対応するエントリの追い出しの優先順序を低く、換言すれば、追い出されにくくなるように設定する。これによって、ＮＩＰローカルメモリにあるエントリとＤＩＰローカルメモリにあるエントリの重複が起こる確率を少なくすることができる。なお、どのエントリを追い出すかを決定するには一般にＬＲＵアルゴリズムが使用される。

【００６１】ＳＴＥＰ９２において、追い出し通知でないと判断された場合、ＮＦＳ要求であるか否かが判断され（ＳＴＥＰ９６）、ＮＦＳ要求である場合にはＮＦＳ（Network File System ）処理を行う（ＳＴＥＰ９７）。一方、ＮＦＳでないと判断された場合には、ＤＩＰローカルメモリのキャッシュを検索し（ＳＴＥＰ９８）、ヒットしたか否かを判断する（ＳＴＥＰ９９）。ここで、ヒットした場合には、ＴＴＬのチェックを行い（ＳＴＥＰ９１０）、有効期限内であれば（ＳＴＥＰ９１１）、キャッシュからデータを読み出して、要求もとのＮＩＰに応答する（ＳＴＥＰ９１２）。

【００６２】キャッシュの検索がヒットしなかった場合、または有効期限内でないと判断された場合には、キャッシュから追い出すエントリを決定し、キャッシュエントリを空け（ＳＴＥＰ９１３）、ストレージデバイス１４より読み出して（ＳＴＥＰ９１４）、該空いたエントリに読み出されたデータを書き込むとともに、要求もとのＮＩＰにデータを送出する（ＳＴＥＰ９１５）。

【００６３】このように、ファイル（データ）をネットワークに送出する場合は、ＮＩＰ１１およびＤＩＰ１３が協調して動作し、第１のバッファメモリ手段であるＮＩＰローカルメモリ２２および第２のバッファメモリ手段であるＤＩＰローカルメモリ７２をそれぞれキャッシュとして利用し、該キャッシュにデータが存在しない場合には、ストレージデバイス１４からデータを読み出している。なお、ストレージデバイス１４からの読み出しは、ＤＩＰのファイルシステム（例えば、プロセッサ７１によるプログラムの実行により実現される）によって行われる。

13

【００６４】次に、本発明に係るサーバへの書き込み動作について説明する。書き込みは、ネットワークから行われる。ネットワークからの書き込み要求は、ＮＩＰ１１が適切なＤＩＰ１３に要求を転送する。適切なＤＩＰを選択するのはＮＩＰ１１が上述したファイル名によるハッシングを利用して行われる。ＮＩＰから要求を与えられたＤＩＰ１３は、ストレージデバイスに対して要求されたデータの書き込みを行う。また、ネットワークからＤＩＰ１３に接続されたストレージデバイス１４にはＮＦＳで読み出し、書き込みができるようにしておく。この機能を実現するために、ＮＩＰ１１は、ネットワークからのＮＦＳ通信をＤＩＰ１３に中継する処理が行われる。

【００６５】ネットワークからＮＦＳでＤＩＰ１３に接続されたストレージデバイス１４に読み書きができることで、ネットワーク上のホストマシンから見てサーバ装置のストレージデバイス１４がリモートファイルシステムとして実現される。ホストマシンからリモートファイルシステムとしてみえることで、ホストマシン上にある従来のソフトウェアを有効に活用することが可能である。本実施例ではＮＦＳをファイル書き込みのプロトコルとして説明したが、他のネットワークプロトコルを使用してもよい。

【００６６】

【発明の効果】以上説明したように、本発明によれば、ストレージデバイスに蓄積されたファイルを効率よくネットワークに送り出すことができる。また、階層化されたキャッシュ管理方式により、高いヒット率を得ることができる。さらに、データをネットワークに送出するとき必要になるパリティ計算をデータの転送中に行うことでプロセッサの負荷を減らすことができる。

14

【図面の簡単な説明】

【図１】　本発明の実施形態に係るシステムの概略構成およびサーバ装置の構成を示す図。

【図２】　ネットワークインターフェースの構成を示す図。

【図３】　キャッシュエントリの構造を示す図。

【図４】　ＮＩＰのパラレルリンクインタフェースの構成を示す図。

【図５】　ＮＩＰにネットワークから要求がきた場合の処理を説明するための図。

【図６】　ＮＩＰがＤＩＰからの応答を受け付けた場合の処理を説明するための図。

【図７】　ＤＩＰの構成を示す図。

【図８】　ＤＩＰのパラレルリンクインタフェースの構成を示す図。

【図９】　ＤＩＰの動作を説明するための図。

【符号の説明】

１…サーバ装置

２…ネットワーク

３…ホストマシン

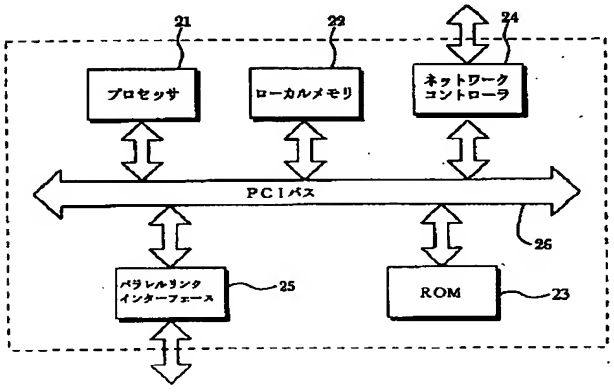
１１…ネットワークインターフェースプロセッサ（ＮＩＰ）

１２…パラレルリンク（バス）

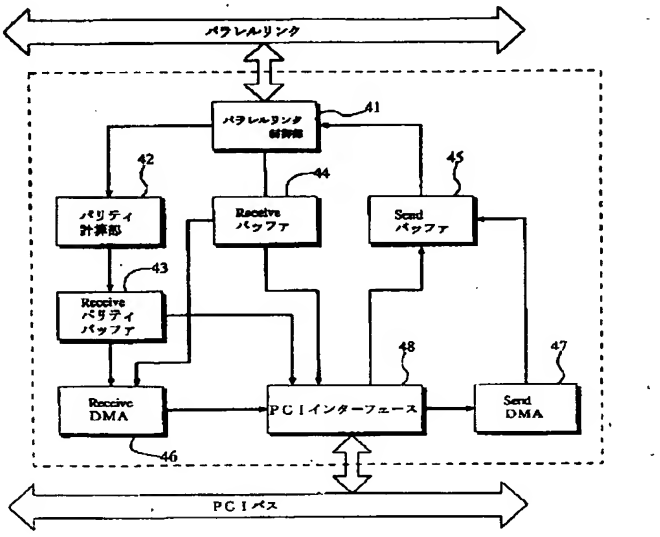
１３…ディスクインターフェースプロセッサ（ＤＩＰ）

１４…ストレージデバイス

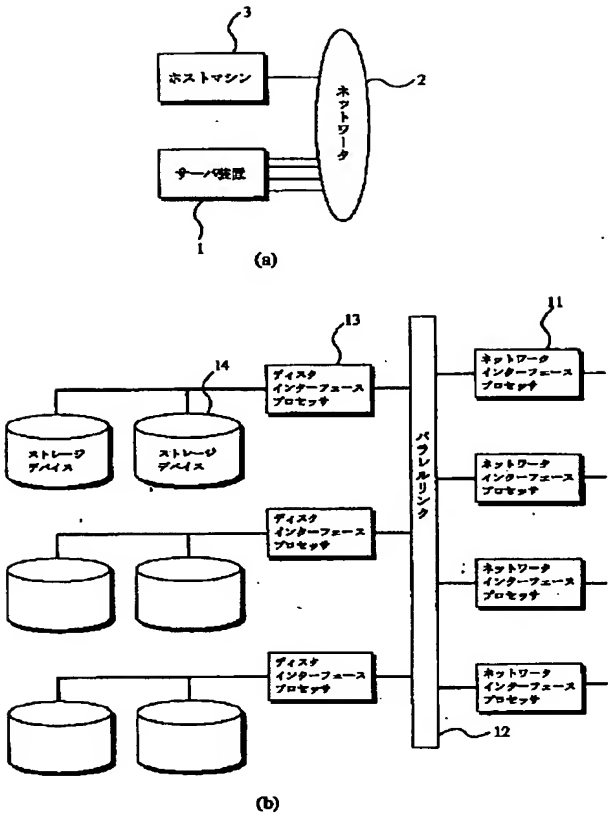
【図２】



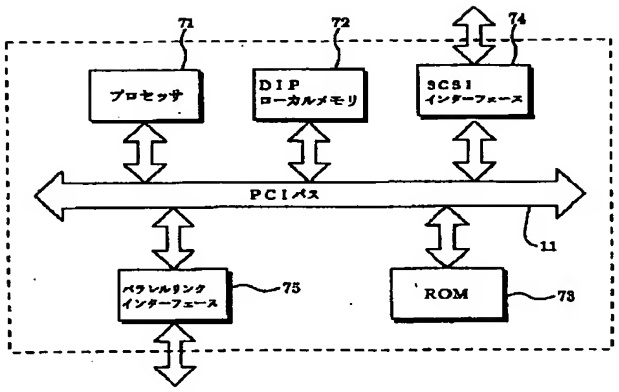
【図４】



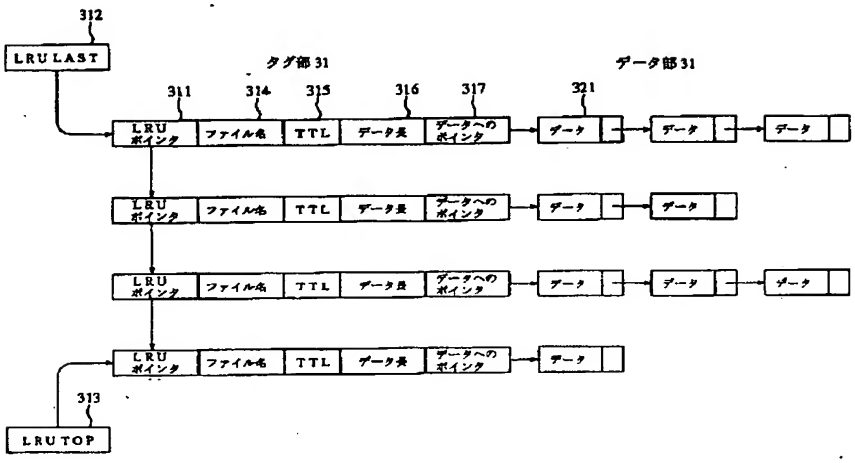
【図１】


3
ホストマシン
2
ネットワーク
サーバ装置
1
(a)
13
11
14
ディスクインターフェースプロセッサ
ネットワークインターフェースプロセッサ
ストレージデバイス
パラレルリンク
12
(b)

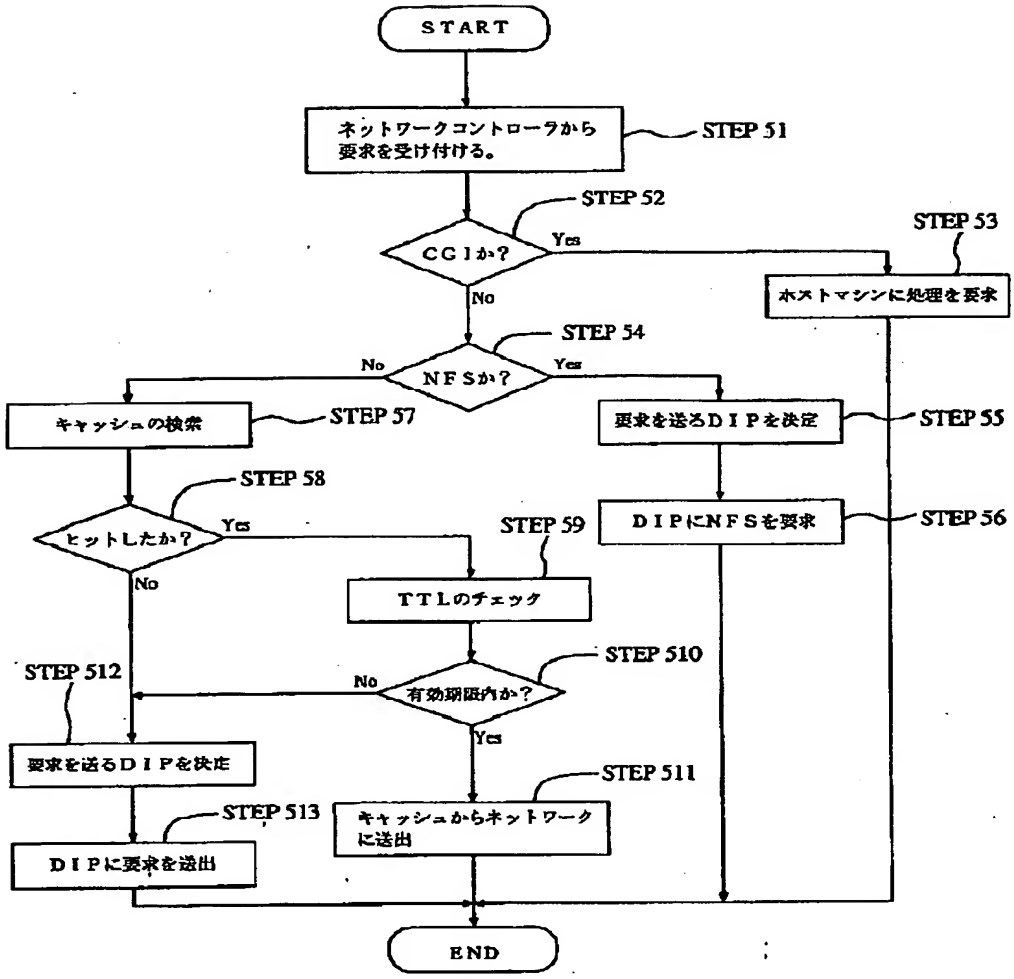
【図７】


71
72
74
プロセッサ
DIP
ローカルメモリ
SCSI
インターフェース
PCIバス
11
パラレルリンク
インターフェース
75
ROM
78

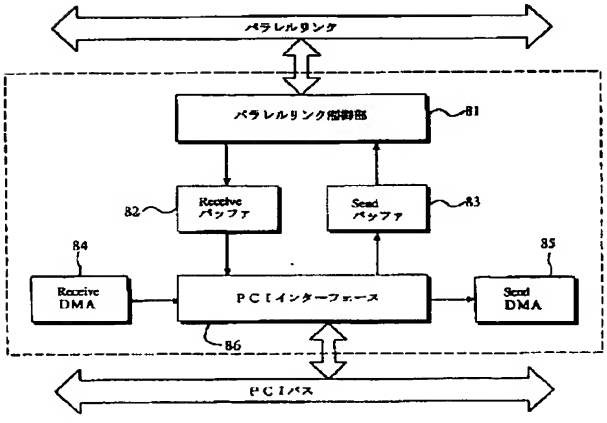
【図３】


312
LRU LAST
タグ部31
データ部31
311
314
315
316
317
321
LRU
ポインタ
ファイル名
TTL
データ長
データへの
ポインタ
データ
313
LRU TOP

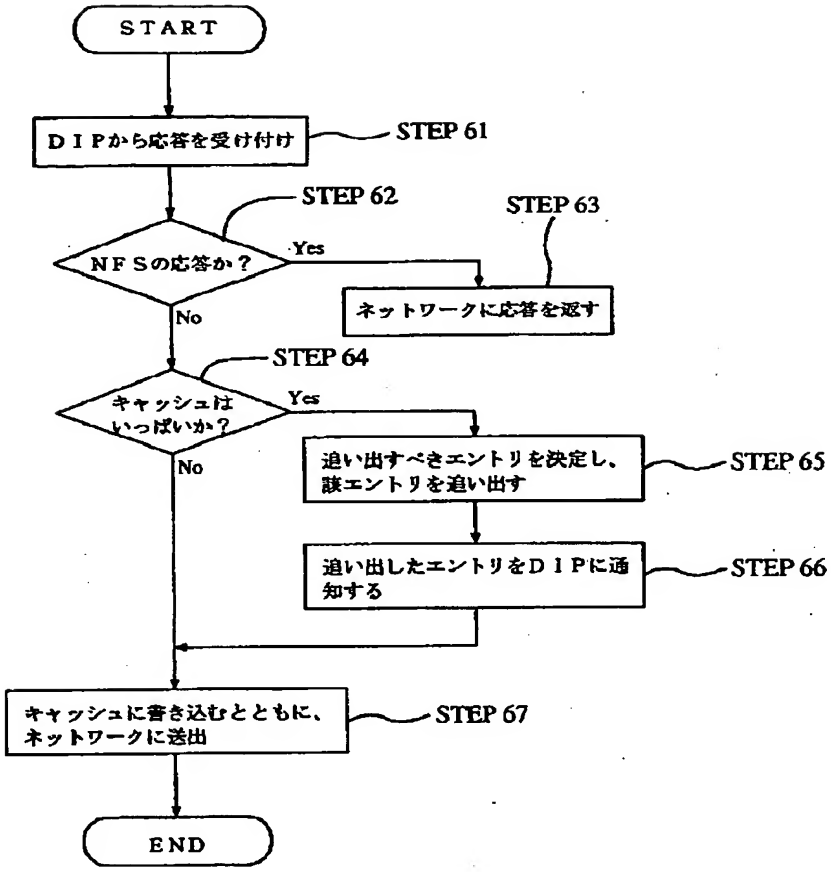
【図５】


START
ネットワークコントローラから要求を受け付ける。
STEP 51
STEP 52
CG1か?
Yes
No
STEP 53
ホストマシンに処理を要求
STEP 54
NFSか?
No
Yes
キャッシュの検索
STEP 57
要求を送るDIPを決定
STEP 55
STEP 58
ヒットしたか?
Yes
No
STEP 59
DIPにNFSを要求
STEP 56
TTLのチェック
STEP 510
有効期限内か?
No
Yes
STEP 512
要求を送るDIPを決定
STEP 511
キャッシュからネットワークに送出
STEP 513
DIPに要求を送出
END

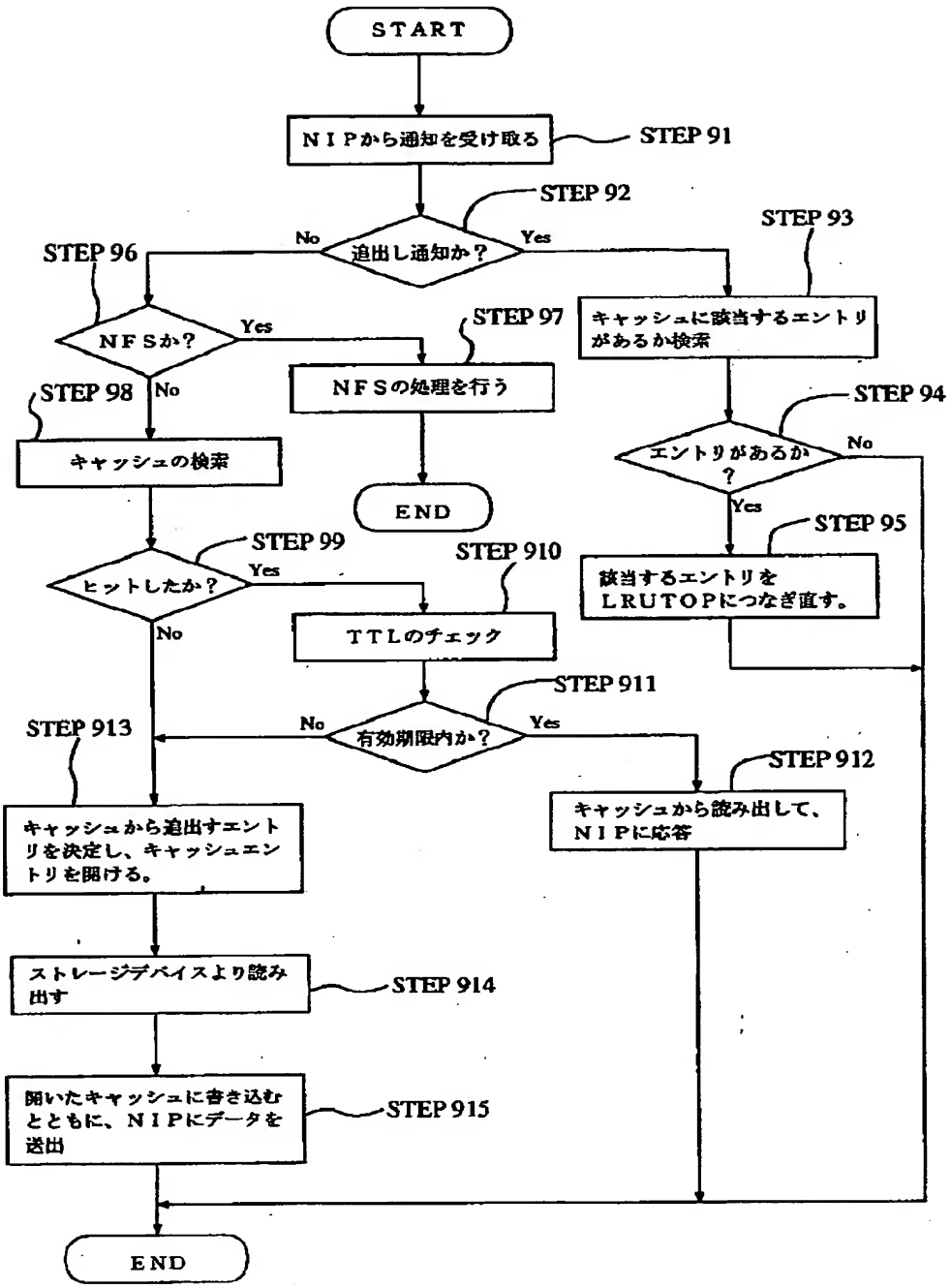
【図８】


パラレルリンク
パラレルリンク制御部
81
Receive バッファ
82
Send バッファ
83
84
85
Receive DMA
PCIインターフェース
Send DMA
86
PCIバス

【図６】


START
ＤＩＰから応答を受け付け
STEP 61
STEP 62
ＮＦＳの応答か？
Yes
STEP 63
ネットワークに応答を返す
No
STEP 64
キャッシュは
いっぱいか？
Yes
No
追い出すべきエントリを決定し、
該エントリを追い出す
STEP 65
追い出したエントリをＤＩＰに通
知する
STEP 66
キャッシュに書き込むとともに、
ネットワークに送出
STEP 67
END

【図９】



フロントページの続き

(72)発明者　前田　誠司
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内